WHOI-95-04

Woods Hole
Oceanographic
Institution

WOODS HOLE OCEANOGRAPHIC INSTITUTION
1930

Report on the Acoustic Network Arctic Deployment, March 1994

by

Mark Johnson, David Herold, Josko Catipovic

March, 1995

Technical Report

Funding was provided by the Office of Naval Research under Contract No. N00014-93-1-0988 and the Advanced Research Projects Agency under Contract Nos. MDA972-91-J-1004 and MDA972-93-1-0019.

Approved for public release; distribution unlimited.

WHOI-95-04

Report on the Acoustic Network Arctic Deployment, March 1994

by

Mark Johnson, David Herold, Josko Catipovic

Woods Hole Oceanographic Institution
Woods Hole, Massachusetts 02543

March 1995

Technical Report

Funding was provided by the Office of Naval Research under Contract No. N00014-93-1-0988 and the Advanced Research Projects Agency under Contract Nos. MDA972-91-J-1004 and MDA972-93-1-0019.

Approved for public release; distribution unlimited.

Approved for Distribution:

George V. Frisk

George V. Frisk, Chair
Department of Applied Ocean Physics and Engineering

Report on the Acoustic Network Arctic Deployment, March 1994

Mark Johnson, David Herold, Josko Catipovic

Woods Hole Oceanographic Institution

1.0 Summary

This report describes the March 1994 Arctic deployment undertaken by the Acoustic Telemetry Group of WHOI. The deployment was a part of the 1994 Sea Ice Mechanics Initiative (SIMI) project and was based at the west SIMI camp, approximately 150 Nautical miles north-east of Prudhoe Bay, Alaska. The goal of the deployment was to install a network of six high-performance acoustic modems, developed at WHOI, and to obtain a data set demonstrating the communications and acoustic monitoring capabilities of the network.

The six modems in the network were deployed over an area of 22 square km and communicated via radio Ethernet with a computer at the SIMI camp. Each modem had a global positioning system, an acoustic source and an 8 element receiving array. The network was operated in a round-robin broadcast mode (i.e., each modem in turn transmitted a packet of data while the others received). The transmissions were 5000 bits-per-second QPSK with a 15kHz carrier. An extensive data set including raw acoustic data, source localization information, and modem position was collected during the deployment.

An additional function of the acoustic network was to communicate with, and track, the Odyssey, an autonomous underwater vehicle operated by the MIT group at the SIMI camp. To this end, the Odyssey was equipped with a Datasonics modem configured for periodic QPSK transmission to the network. A data set was obtained from which both the up-link communication and localization capabilities of the network can be determined.

2.0 Introduction

The Arctic deployment took place from March 21 to March 30, 1994, at the west Sea Ice Mechanics Initiative (SIMI) camp which was situated approximately 150 Nautical miles north-east of Prudhoe Bay, Alaska. The WHOI Advanced Engineering Laboratory (AEL) personnel involved in the deployment were Josko Catipovic, Ed Denton, David Herold, and Mark Johnson.

The aim of the deployment was to install and operate a network of six high-performance acoustic modems developed by AEL. As shown conceptually in Fig. 1, each modem was equipped with an acoustic source, an acoustic receiving array, and a radio Ethernet link to a Sun SPARC computer at the base camp. The base camp computer was used both to collect data from the modems and to control the modem functions.

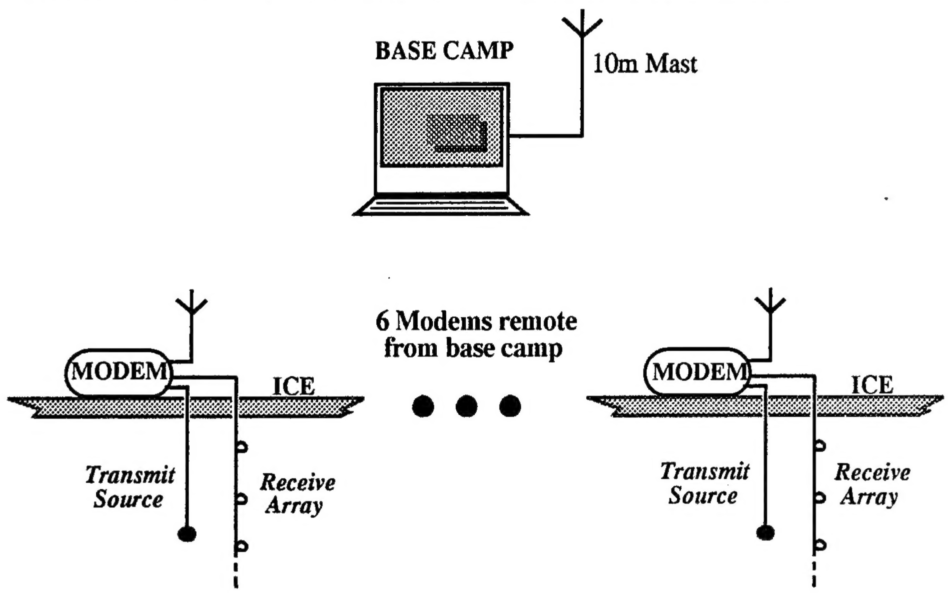

Fig. 1 Conceptual diagram of the Arctic deployment.

The modem hardware shown in Fig. 2, comprises a personal-computer (PC) system controller, two high-speed digital signal processors (DSPs), analog-to-digital converters (ADCs), digital-to-analog converters (DACs), and a global positioning system (GPS) together with a number of diagnostic peripherals. The unit is battery-powered and incorporates both power and temperature sensing and control circuitry. Further details of the modem hardware and software may be found in. From the viewpoint of the network operator, each modem is associated with a DOS window on the base camp computer, from which all aspects of the modem can be directly controlled.

The modem network deployed in the Arctic was configured to operate in a broadcast mode in which one modem would transmit and the remaining modems in the network would receive the transmitted data. The transmissions used quadrature phase shift keying (QPSK) modulation with a carrier frequency of 15kHz and a data rate of 5000bits-per-second (bps). Following a transmission, each modem logged the received signal (demodulated and decimated to reduce storage requirements) and its precise arrival time

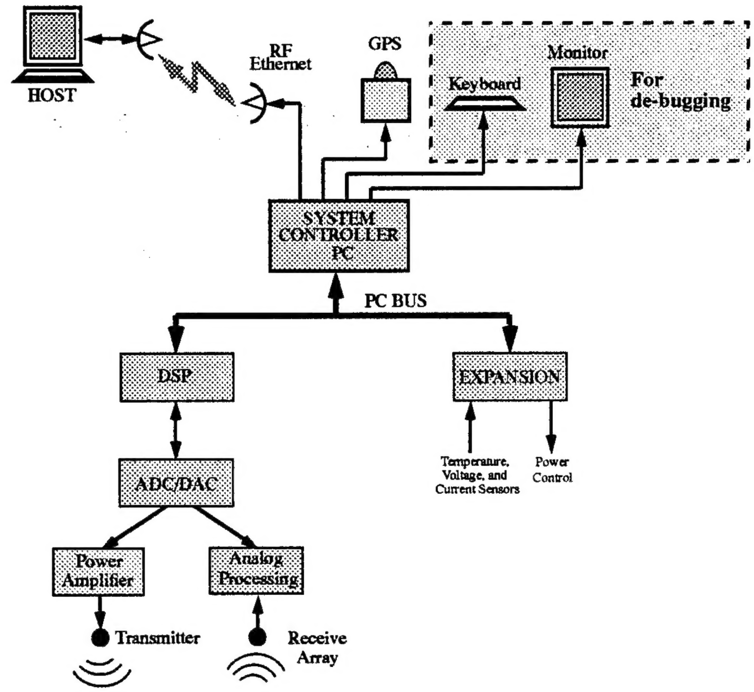

Fig. 2 System Diagram of Acoustic Modem with associated peripherals

together with diagnostic data regarding the power consumption, temperature, and GPS position of the modem. A report was also available from each modem indicating the performance of the receiver algorithm in decoding the transmitted signal. Experimental results were collected by the base camp computer and displayed using Matlab. The parameters used in each modem for transmission and reception (e.g. source level, synchronization codes, equalizer settings, etc.), could be readily accessed and changed from the base camp computer.

An additional function of the Arctic network was to demonstrate communication with, and tracking of, the Odyssey, an autonomous underwater vehicle (AUV) operated by the MIT group at the SIMI camp. The Datasonics modem used on the Odyssey was configured solely for periodic transmission preventing two-way communication with the AUV. However, data obtained with the network acting as a receiver demonstrates both the up-link communication and localization capabilities of the network.

In the following section, some aspects of the design of the communication network are discussed. A summary of the data-logging experiments performed with the network is then given in Section 4 followed by a description of the logged data formats and structures in Section 5. Complete experimental descriptions are given in Section 6 and Matlab and C programs used to extract information from the acquired data base are provided in Section 7. The report concludes with references and appendices.

3.0 Communication System Description

In the following sub-sections details of the transmitted signals and the transmit and receive transducers are given.

3.1 Acoustic Transducers

The vertical receive array used with each modem comprised 8 omni-directional transducers each separated by 1m. The transducers were numbered 1-8 with element 1 being the lowest. By convention, element 1 was used for single channel reception while elements 1, 3, 5, and 7 were used for four channel reception (i.e., an effective inter-element spacing of 2m).

The transmit sources were Datasonics AT-18DT 15kHz directional transducers with a beamwidth of approximately +/- 15 degrees from the horizontal. The usable frequency band of these transducers is between 8kHz and 22kHz.

Modem sites were chosen with low ice thickness (i.e., <3m) and the sources and hydrophone arrays were lowered through holes in the ice to a depth of about 100m.

3.2 Transmitted Signal Description

All transmissions were 5000bps QPSK centered at 15.151kHz. The transmitted packet consists of three sections: first, a 13 chip Barker code sequence was transmitted at a rate of 1 chip per symbol to enable synchronization at the receiver. This was followed by a period of silence (typically of length 20 symbols), required to enhance receiver Barker detection in multipath environments. The third section of the transmitted packet contained the QPSK-modulated data. The number of data symbols in a packet was typically 1000, 2000, or 10000 depending upon the number of receiver channels used.

The settings used by the transmitter are contained in the txm.m file, a text file used directly by the DSPs in each modem. An example txm.m file is given below:

```
% txm.m for 10000 symbol QPSK packet transmission

GPS_EXIST=1                          % Enables GPS-based timing
time(1)                              % Starts GPS pulse-per-second timer
file = 'ltrain.bin'                  % Binary data file to be transmitted
map = [1+j;-1-j;-1+j;1-j];           % Symbol mapping for QPSK
code = [-1;1;-1;1;-1;-1;1;1;-1;-1;-1;-1;-1] ;% 13 chip Barker code
txf = [1;1;1;1;1;1;1;1;1;1;1;1];
nullf = [0;0;0;0;0;0;0;0;0;0;0;0];
txf = [txf;txf;nullf;nullf];         % Passband transmit filter
rate = [2500;15151;60000;0.5;0;20;10000;200;1.5;0.0];% Options for transmitter
transmit(file,map,code,txf,rate)     % Transmit a packet
quit                                 % Return to DOS shell
```

The numbers in the rate vector defined above control a number of aspects of transmitter performance as follows:.

Option:	Units	Place in rate vector
data-rate	symbols/sec	1
carrier frequency	Hz	2
sampling-rate	Hz	3
amplitude	-	4
output channel	-	5
silence period length	symbols	6
packet length	symbols	7
block size (internal)	symbols	8
Barker code magnitude (real and imag.)	-	9,10

NOTE: No pulse shaping or transmit filtering was applied.

3.3 Receiver Description

The settings used by the receiver are contained in the rxd1.m file, a text file used directly by the DSPs in each modem. An example rxd1.m file is given below:

```
% rxd1.m for 1000 symbol QPSK packet reception

file = 'out.da'                                  % data file
map = [1+j;-1-j;-1+j;1-j];                       % qpsk constellation
code = [-1;-1;1;1;-1;-1;1;1;-1;-1;-1;-1
     1;1;1;1;-1;-1;-1;-1;-1;-1;-1;-1;-1;-1] ;  % barker sequence
rxf = [-0.0013;-0.0011;-0.0008;0.0;0.0019;0.0053;0.0107] ;
rxf = [rxf;0.0182;0.0277;0.0388;0.0508;0.0626;0.0734;0.0819];
rxf = [rxf;0.0874;0.0893;0.0874;0.0819;0.0734;0.0626;0.0508] ;
rxf = [rxf;0.0388;0.0277;0.0182;0.0107;0.0053;0.0019;0.0];
rxf = [rxf;-0.0008;-0.0011;-0.0013] ;        % receive filter settings
%
N = [30,1;30,0] ;                                % equalizer settings
pll = [0.0011,-0.001,0;1,-2,1] ;                 % pll settings
rate = [2500;15151;60000;35;1000;0.99;1.0;2;0;0.001;0.05;100;10;20;350;50.0] ;
receive(file,map,code,rxf,'qtrain.dat',N,pll,rate);   % receive a packet
quit                                             % exit
```

The key features of this file are: 1) the receive filter (rxf) which is a 31 tap linear-phase windowed filter, 2) N which tells the number of feedforward and feedback parameters in the equalizer as well as the channel number the data was received on (N(ff parameters, channel; ff parameters ,channel; ... ;fb parameters, fb channel)). The critical components in the rate vector are: 2500 symbol rate, 15151 carrier frequency, 60000 sample frequency, 1000 packet length in symbols, and 2 which is the inverse of the fractional spacing and determines the data rate of the logged sampled data (i.e. 2 x 2500 = 5000 samples per second). The actual symboling rate is 1/24 of the sample rate which is

determined by 2e6/X, where X is an integer. In this case X = 33 and the sampling rate is 60606 Hz, thus the actual symbol rate is 60606/24 = 2525 symbols per second.

4.0 Experiment Summary

4.1 Summary of Data Capture Experiments

A total of 9 data capture experiments were performed between March 26 and March 30 yielding about 500Mbytes of data. The experiments are summarized below and described in more detail in Section 6. The experiment code given for each experiment provides a cross-reference with the log book of the Arctic deployment. All times given in this report are local (central Alaskan) time which is 9 hours behind GMT.

Expt. 1: *March 26, 8:10* Round-robin transmissions in a four node network; 10K symbol packets (also some decoded 1K symbol packets); single channel reception. This data set was taken the morning after a substantial lead opened between Sites 3,2 and Sites 1,6. Experiment code: *rr*.

Expt. 2: *March 27, 14:03* Transmissions from a Datasonics modem on the MIT Odyssey received by a five node network; 1K symbol packets; single channel reception. Experiment code: *mit1*.

Expt. 3: *March 27, 22:38* Round-robin transmissions in a four node network with a change of transmitter every hour; 5K symbol packets; single channel reception. Experiment code: *on27_3*.

Expt. 4: *March 28, 10:12* Round-robin transmissions in a three node network with a change of transmitter every hour; 10K symbol packets; single channel reception. The Datasonics modem on the MIT Odyssey was transmitting during the final hour of this experiment. Experiment code: *on28_3*.

Expt. 5: *March 28, 22:00* Transmissions from a Datasonics modem on the MIT Odyssey received by a six node network; 1K symbol packets; four channel reception. Experiment code: *mit2*.

Expt. 6: *March 28, 22:22* Round-robin transmissions in a six node network with a change of transmitter every hour; 2K symbol packets; four channel reception. Experiment code: *on28_3a*.

Expt. 7: *March 29, 9:57* Round-robin transmissions in a six node network with a change of transmitter every hour; 2K symbol packets; four channel reception. Experiment code: *on29_3*.

Expt. 8: *March 29, 22:40* Round-robin transmissions in a six node network with a change of transmitter every hour; 10K symbol packets; single channel reception. Experiment code: *on29_3a*.

Expt. 9: *March 30, 10:05* Simultaneous transmissions from two modems to remaining modems in a six node network; 2.5K symbol packets; four channel reception. Experiment code: *collide*.

4.2 Modem Site and Package Naming Convention

Six modem sites were selected for the network nodes covering an area of approximately 22 square km roughly centered on the base camp. Acoustic transducers were permanently installed at each site. However, the modem electronics packages (contained in insulated boxes) were swapped from one site to another on several occasions as service was required. To draw a distinction between the modem sites and the modem packages, the sites were numbered 1-6 and the packages were given the names: bambi, bashful, doc, dopey, gary, wally. The data collected from each modem is stored in a directory of the data base with the appropriate package name.

The location of the modem packages throughout the 9 data capture experiments is summarized in the following table. A blank entry indicates that no modem package was present at the site concerned.

TABLE 1. Site Table (Experiment Configuration)

Experiment	Site 1	Site 2	Site 3	Site 4	Site 5	Site 6
1 (rr)	dopey	bashful	doc		gary	
2 (mit1)	dopey	bashful	doc	gary	bambi	
3 (on27_3)	dopey	bashful		gary	bambi	
4 (on28_3)	dopey	bashful	doc	gary	bambi	
5 (mit2)	dopey	bashful	wally	gary	bambi	doc
6 (on28_3a)	dopey	bashful	wally	gary	bambi	doc
7 (on29_3)	dopey	bashful	wally	gary	bambi	doc
8 (on29_3a)	dopey	bashful	wally	gary	bambi	doc
9 (collide)	dopey	bashful	wally	gary	bambi	doc

4.3 Sources of Error in the Data Set

A number of error and interference sources are known to have influenced the experimental data set in terms of the received acoustic signals, the reception times, the GPS positions, and the voltage and temperature measurements. These are summarized in the following.

i) Received Acoustic Signals: Two sources of noise/interference were observed during the acoustic data capture. First, the ambient noise level fluctuated widely depending upon the wind speed. At high wind speeds, considerable impact noise from blown snow and ice particles was transmitted from the ice into the water. This noise source is known to have a substantial component in the frequency range used for acoustic communications (10kHz - 20kHz). An hourly log of the wind speeds during the deployment has

been obtained from PMEL at NOAA (Appendix B) and an attempt will be made to correlate the wind speed with the ambient noise levels in the captured acoustic data.

The second interference source affecting the acoustic data was the acoustic navigation beacons used by the MIT group. Certain of these beacons generated 5ms pulses with high energy in the communications frequency band. The pulses occurred every second.

ii) Reception Timing: An error in the received packet timing software in the acoustic receiver resulted in an occasional timing ambiguity. The magnitude of the ambiguity is exactly 1 second and can be readily corrected in the data without loss of confidence. It should also be noted that if the timing figure of merit (TFOM) of the GPS is >7 or F, the reception timing should be disregarded.

iii) GPS: The accuracy of the GPS 'xyz' position is within 100m due to the DOD selective availability randomization.

iv) Temperature and Voltage: The temperature and voltage readings logged during Expts. 3,4,6,7,8 are to be disregarded. These measurements were taken with an essential power supply disabled and so are inaccurate.

5.0 Result Formats

5.1 Data File Format

Data collected from each node are stored in files which are named with the following conventions: out<minute>.<da|da1>, where: out is the prefix for all output data files; <minute> is the minute of the day that the data file was created (computed from GMT); and <da|da1> is the file extension. For example, in a particular experiment subdirectory for a given node there might be two files named: out889.da and out889.da1. Due to certain conditions (copy times over the RF link and security circuit time-out period) data had to be transmitted to the base camp as two separate files. Therefore, a file with a *da1* extension is a continuation of the file with the same name and a *da* extension.

Critical information about the usage of this data is contained in the header of the.da files. This information has the following format:

1234 53352.000000
    1234: Arctic Data Magic Number
    53352.0000000: Nearest Second of Day to detect
17120 17528
    17120: Index into data for start of packet
    17528: Index into data for stop of packet
4 3
    4: Number of channels of data
    3: Number of channels on which a detect occurred

There is one entry in the following format for every channel on which data was collected.

1 -0.693600
    1<0>: 1 is detect on this channel, 0 no detect
    -0.693600: Offset in seconds from Nearest second of day to a detect to the actual detection.
1 -0.693600
1 -0.684200
0 -0.693600

The data is stored in two columns, the left hand column is the real component of the data and the right hand column is the imaginary. Data samples are stored in ascending order from earliest to latest. Multi-channel data is stored in the following sequence:

Channel 1, Sample 1
Channel 2, Sample 1
Channel 3, Sample 1
Channel 4, Sample 1
Channel 1, Sample 2
Channel 2, Sample 2
:

5.2 STATUS.LOG File Format(s)

There are two STATUS.LOG file formats as indicated below. Both contain time information, latitude, longitude, temperature measurements, battery voltage and current and an indicator as to whether the modem was transmitting or receiving at the time.

Format A is used in experiments 1 through 4, and Format B is used in experiments 6 through 8. Experiments 5 and 9 have no STATUS.LOG files.

5.2.1 Format A

HH:MM:SS TFOM = X DDMM.HH,N,DDDMM.HH
T0 SDT0 T1 SDT1 T2 SDT2 T3 SDT3 BATT CURRENT
txlrx one
:

HH:MM:SS	GMT Hours (HH): Minutes (MM): Seconds (SS)
TFOM = X	Timing Figure of Merit: X is an integer between 1 and 15 which indicates the accuracy of the timing pulse from the GPS (See Table).
DDMM.HH,N,DDDMM.HH	Latitude: Degrees (DD) Minutes (MM). Hundredths of Minutes (HH), North (N), Longitude: Degrees (DDD) Minutes (MM). Hundredths of Minutes (HH)
T0 SDT0...T3 SDT3*	(T0-T3) Temperature for channels 0-3 in degrees Celsius. (SDT0-SDT3) Standard Deviation of Temperature Measurement for channels 0-3 in degrees Celsius.
BATT	Battery voltage (volts).
CURRENT*	Supply Current (amps).
txlrx one	Mode indicator: (tx) transmitter mode, (rx) receiver one: indicates a single transmission or reception.

*See individual experiment description for details.

5.2.2 Format B

HH:MM:SS TFOM = X DDMM.HH,N,DDDMM.HH
T0 SDT0 T1 SDT1 T2 SDT2 T3 SDT3 BATT CURRENT
XXXX YYYY ZZZZ
txlrx one
:

HH:MM:SS	GMT Hours (HH): Minutes (MM): Seconds (SS)
TFOM = X	Timing Figure of Merit: X is an integer between 1 and 15 which indicates the accuracy of the timing pulse from the GPS (See Table).
DDMM.HH,N,DDDMM.HH	Latitude: Degrees (DD) Minutes (MM). Hundredths of Minutes (HH), North (N), Longitude: Degrees (DDD) Minutes (MM). Hundredths of Minutes (HH)
T0 SDT0...T3 SDT3*	(T0-T3) Temperature for channels 0-3 in degrees

	Celsius. (SDT0-SDT3) Standard Deviation of Temperature Measurement for channels 0-3 in degrees Celsius.
BATT	Battery voltage (volts).
CURRENT*	Supply Current (amps).
txlrx one	Mode indicator: (tx) transmitter mode, (rx) receiver one: indicates a single transmission or reception.
XXXX YYYY ZZZZ	X, Y and Z coordinates of modem in meters.

*See individual experiment description for details.

TABLE 2. Timing Figure of Merit (TFOM)

TFOM Value	Expected Time Error (min)	Expected Time Error (max)
1	0nS	1nS
2	1nS	10nS
3	10nS	100nS
4	100nS	1uS
5	1uS	10uS
6	10uS	100uS
7	100uS	1mS
8	1mS	10mS
9	10mS	100mS
10-14	NOT USED	NOT USED
15	No information	Available

6.0 Experiment Descriptions

Details of the 9 data-logging experiments performed are given in the following sub-sections.

6.1 Experiment 1 *(rr)*

Description:	Round-robin of transmissions from each modem to remaining modems.
Data purpose:	Analysis of network connectivity, modem localization, and comparative performance of the network paths.
Started:	8:10, March 26
Duration:	approx. 3 hours
Number of nodes:	4
Number of channels:	1
Packet length:	1000 symbols
Raw data:	/data0/marky/arctic/rr
Processed data:	/data0/marky/arctic/expt1

TABLE 3. GPS positions, Expt 1.

Node	GPS
bashful	72:56.50N, 148:26.95W
doc	72:57.04N, 148:29.91W
dopey	72:57.97N, 148:31.21W
gary	72:59.03N, 148:28.32W

6.2 Experiment 2 *(mit1)*

Description:	Receptions from Datasonics modem on the Odyssey (Odyssey approx. 8m below ice surface).
Data purpose:	Analysis of AUV up-link performance, AUV localization.
Started:	14:03, March 27
Duration:	3 hrs 20 mins
Number of nodes:	5
Number of channels:	1
Packet length:	1000 symbols
Raw data:	/data0/marky/arctic/mit1
Processed data:	/data0/marky/arctic/expt2

TABLE 4. GPS positions, Expt 2.

Node	GPS
bambi	72:59.42N, 148:00.39W
bashful	72:56.86N, 147:58.87W
doc	72:57.39N, 148:01.83W
dopey	72:58.38N, 148:03.32W
gary	72:56.66N, 148:13.33W
Odyssey	TBD

6.3 Experiment 3 *(on27_3)*

Description:	Round-robin with hourly change of transmitting node.
Data purpose:	Analysis of long-term network performance, acoustic channel variability, modem localization, reduced-complexity receiver design.
Started:	22:38, March 27
Duration:	10 hrs, 23 mins
Number of nodes:	4
Number of channels:	1
Packet length:	5000 symbols
Raw data:	/data0/marky/arctic/on27_3
Processed data:	/data0/marky/arctic/expt3

TABLE 5. GPS positions and experiment times, Expt. 3

Node	Start Time	GPS	Stop Time	GPS
bambi	22:38:18	7259.40,N, 14800.37	08:39:59	7258.91,N, 14755.66
bashful	22:38:20	7256.85,N, 14758.84	08:44:16	7256.41,N, 14754.33
dopey	22:38:19	7258.36,N, 14803.29	08:47:02	7257.87,N, 14758.57
gary	22:38:21	7256.65,N, 14813.29	08:43:23	7256.23,N, 14808.88

TABLE 6. Transmission schedule, Expt. 3

Time	Transmitting Node	File Time Stamps
22:38 - 23:00	bambi	420 - 479
23:00 - 0:00	bashful	480 - 539
0:00 - 1:00	gary	540 - 599
1:00 - 2:00	no transmitter	600 - 659
2:00 - 3:00	dopey	660 - 719
3:00 - 4:00	bambi	720 - 779
4:00 - 5:00	bashful	780 - 839
5:00 - 6:00	gary	840 - 899
6:00 - 7:00	no transmitter	900 - 959
7:00 - 8:00	dopey	960 - 1019
8:00 - 8:40	bambi	1020 - 1079

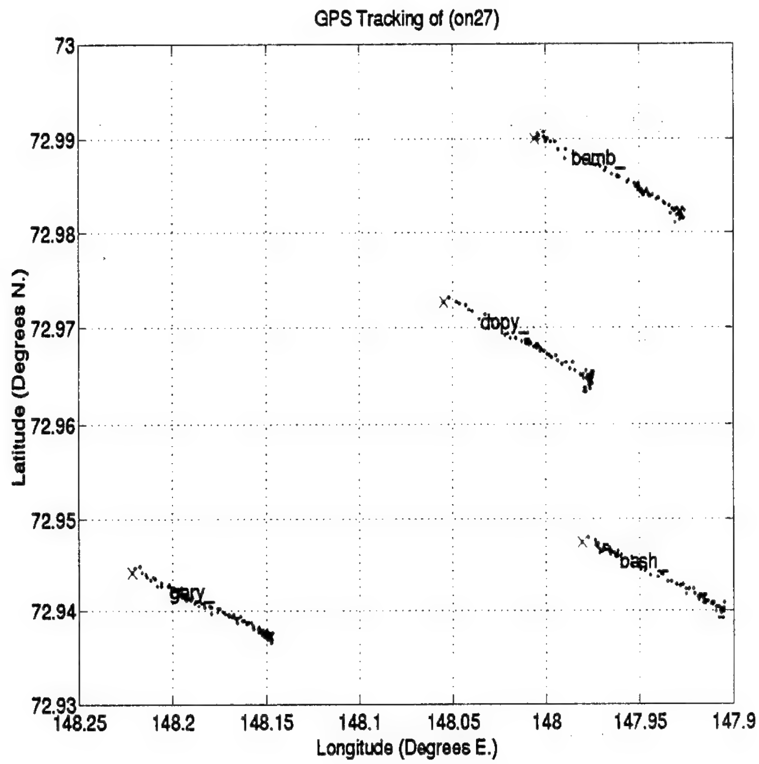

FIGURE 1. GPS tracking, Expt. 3.

6.4 Experiment 4 *(on28_3)*

Description:	Round-robin with hourly change of transmitting node.
Data purpose:	Analysis of long-term network performance, acoustic channel variability, modem localization, reduced-complexity receiver design.
Started:	10:12, March 28
Duration:	5 hrs, 10 mins
Number of nodes:	3
Number of channels:	1
Packet length:	10000 symbols
Raw data:	/data0/marky/arctic/on28_3
Processed data:	/data0/marky/arctic/expt4

TABLE 7. GPS positions and experiment times, Expt. 4.

Node	Start Time	GPS	Stop Time	GPS
bashful	10:12:20	7256.40,N, 14754.35	15:21:58	7256.39,N, 14754.49
dopey	10:12:13	7257.91,N, 14758.41	15:20:33	7257.87,N, 14758.67
gary	10:12:20	7256.26,N, 14808.71	15:20:05	7256.20,N, 14809.03

TABLE 8. Transmission schedule, Expt. 4

Time	Transmitting Node	File Time Stamps
10:12 - 11:00	dopey	420 - 479
11:00 - 12:00	bashful	480 - 539
12:00 - 13:00	gary	540 - 599
13:00 - 14:00	dopey	600 - 659
14:00 - 15:00	bashful	660 - 719
15:00 - 15:20	gary	720 - 779

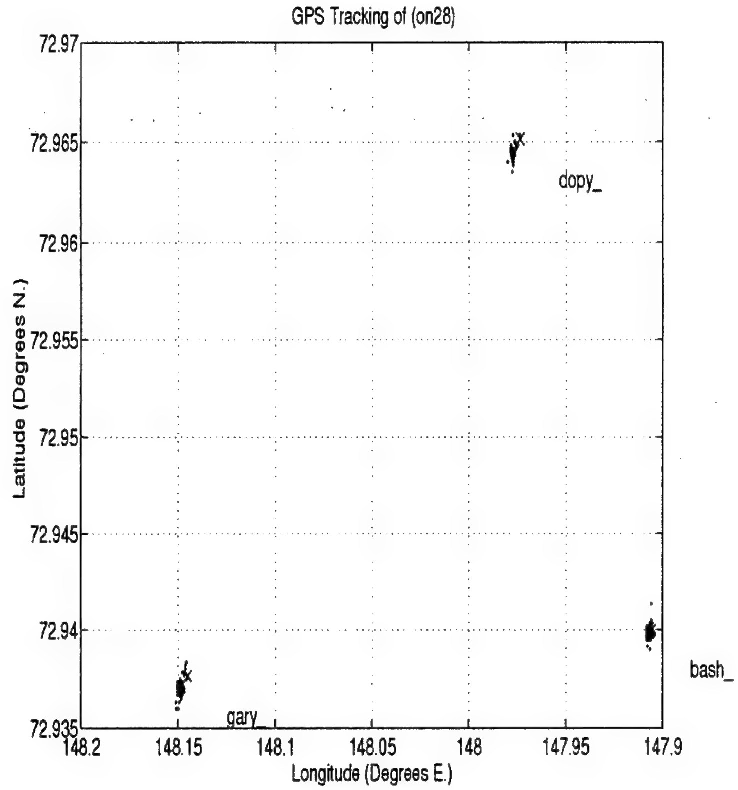

FIGURE 2. GPS tracking, Expt. 4.

6.5 Experiment 5 *(mit2)*

Description:	Receptions from Datasonics modem on the Odyssey (Odyssey approx. 8m below ice surface).
Data purpose:	Analysis of AUV up-link performance, AUV localization, multi-channel (diversity) receiver performance.
Started:	22:00, March 28
Duration:	20 mins
Number of nodes:	6
Number of channels:	4
Packet length:	2000 symbols
Raw data:	/data0/marky/arctic/mit2
Processed data:	/data0/marky/arctic/expt5

TABLE 9. GPS positions, Expt 2.

Node	GPS
bambi	72:58.78N, 147:57.08W
bashful	72:56.28N, 147:55.99W
doc	72:58.68N, 148:00.45W
dopey	72:57.74N,148:00.03W
gary	72:56.10N, 148:10.49W
wally	72:56.83N, 147:58.98W
Odyssey	TBD

6.6 Experiment 6 *(on28_3a)*

Description:	Round-robin with hourly change of transmitting node, multi-channel reception.
Data purpose:	Analysis of long-term network performance, diversity reception, acoustic channel variability, modem localization, reduced-complexity receiver design.
Started:	22:22, March 28
Duration:	9 hrs, 50 mins
Number of nodes:	6
Number of channels:	4
Packet length:	2000 symbols
Raw data:	/data0/marky/arctic/on28_3a
Processed data:	/data0/marky/arctic/expt6

TABLE 10. GPS positions and experiment times, Expt. 6.

Node	Start Time	GPS	Stop Time	GPS
bambi	22:20:27	7258.88,N, 14757.66	08:27:47	7259.73,N, 14807.15
bashful	22:20:35	7256.36,N, 14756.48	08:27:39	7257.26,N, 14806.16
doc	22:20:31	7258.80,N, 14800.99	06:40:39	7259.33,N, 14808.20
dopey	22:20:17	7257.84,N, 14800.58	08:32:36	7258.70,N, 14810.30
gary	22:20:25	7256.18,N, 14811.03	06:43:48	7256.80,N, 14818.44
wally	22:20:22	7256.91,N, 14759.50	08:31:49	7257.80,N, 14809.43

TABLE 11. Transmission schedule, Expt. 6

Time	Transmitting Node	File Time Stamps
22:20 - 23:00	bambi	420 - 479
23:00 - 0:00	dopey	480 - 539
0:00 - 1:00	wally	540 - 599
1:00 - 2:00	gary	600 - 659
2:00 - 3:00	bashful	660 - 719
3:00 - 4:00	doc	720 - 779
4:00 - 5:00	bambi	780 - 839
5:00 - 6:00	dopey	840 - 899
6:00 - 7:00	wally	900 - 959

Time	Transmitting Node	File Time Stamps
7:00 - 8:00	gary	960 - 1019
8:00 - 8:30	bashful	1020 - 1079

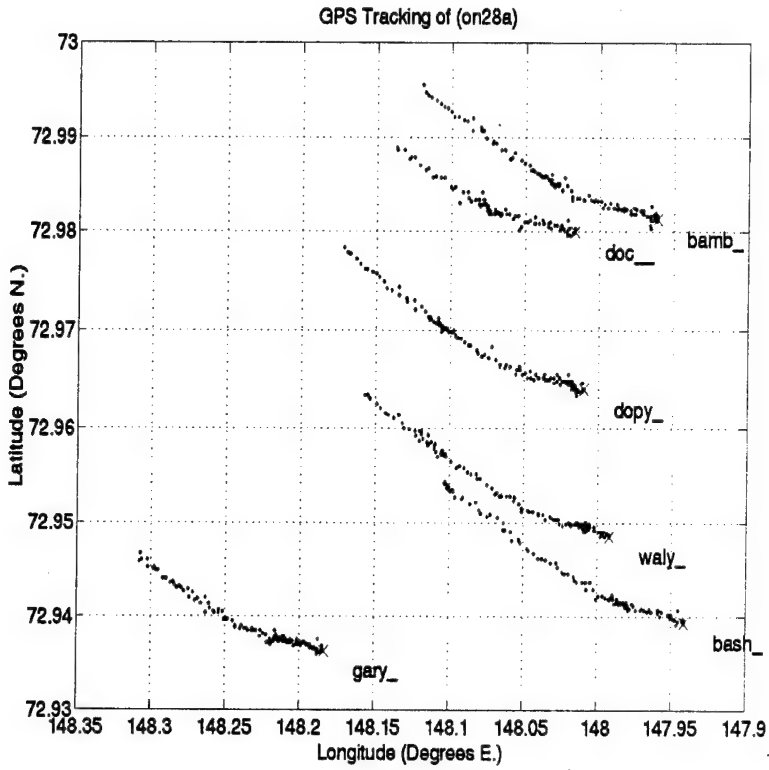

FIGURE 3. GPS tracking, Expt. 6.

6.7 Experiment 7 *(on29_3)*

Description:	Round-robin with hourly change of transmitting node, multi-channel reception.
Data purpose:	Analysis of long-term network performance, diversity equalizer, acoustic channel variability, modem localization, reduced-complexity receiver design.
Started:	9:57, March 29
Duration:	11 hrs, 38 mins
Number of nodes:	6
Number of channels:	4
Packet length:	2000 symbols
Raw data:	/data0/marky/arctic/on29_3
Processed data:	/data0/marky/arctic/expt7

TABLE 12. GPS positions and experiment times, Expt. 7.

Node	Start Time	GPS	Stop Time	GPS
bambi	09:57:39	7300.00,N, 14809.26	21:35:52	7302.81,N, 14825.08
bashful	09:57:51	7257.54,N, 14808.19	21:30:07	7300.30,N, 14823.67
doc	09:57:35	7259.90,N, 14812.62	21:06:30	7302.38,N, 14826.99
dopey	09:57:48	7258.94,N, 14812.20	21:33:24	7301.76,N, 14828.07
gary	09:57:38	7257.34,N, 14822.95	21:13:45	7300.15,N, 14838.29
wally	09:57:53	7258.09,N, 14811.39	21:29:25	7300.89,N, 14826.72

TABLE 13. Transmission schedule, Expt. 7

Time	Transmitting Node	File Time Stamps
9:57 - 10:00	doc	1080 - 1139
10:00 - 11:00	bambi	1140 - 1199
11:00 - 12:00	dopey	1200 - 1259
12:00 - 13:00	wally	1260 - 1319
13:00 - 14:00	gary	1320 - 1379
14:00 - 15:00	bashful	1380 - 1439
15:00 - 16:00	doc	0 - 59
16:00 - 17:00	bambi	60 - 119
17:00 - 18:00	dopey	120 - 179

Time	Transmitting Node	File Time Stamps
18:00 - 19:00	wally	180 - 239
19:00 - 20:00	gary	240 - 299
20:00 - 21:00	bashful	300 - 359
21:00 - 21:29	doc	360 - 419

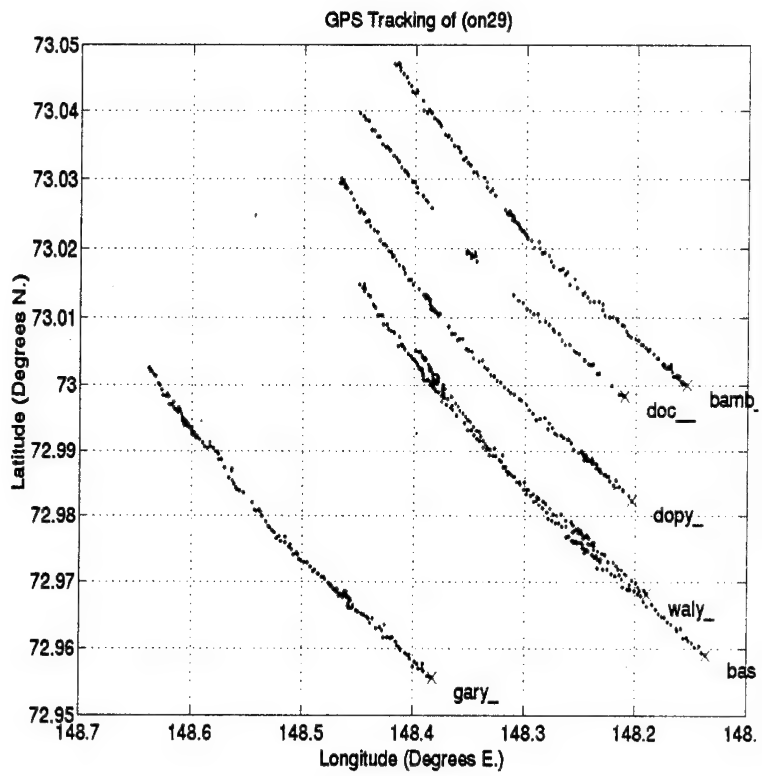

FIGURE 4. GPS tracking, Expt. 7.

6.8 Experiment 8 *(on29_3a)*

Description:	Round-robin with hourly change of transmitting node.
Data purpose:	Analysis of long-term network performance, acoustic channel variability, modem localization, reduced-complexity receiver design.
Started:	22:40, March 29
Duration:	9 hrs, 50 mins
Number of nodes:	6
Number of channels:	1
Packet length:	10000 symbols
Raw data:	/data0/marky/arctic/on29_3a
Processed data:	/data0/marky/arctic/expt8

TABLE 14. GPS positions and experiment times, Expt. 8.

Node	Start Time	GPS	Stop Time	GPS
bambi	22:40:54	7303.18,N, 14826.41	08:09:02	7305.28,N, 14833.38
bashful	22:40:54	7300.67,N, 14825.16	08:08:18	7302.70,N, 14832.04
doc	22:40:49	7303.09,N, 14829.77	08:10:50	7305.19,N, 14836.73
dopey	22:40:52	7302.14,N, 14829.35	08:11:02	7304.24,N, 14836.33
gary	22:40:56	7300.54,N, 14839.76	08:10:35	7302.56,N, 14846.16
wally	22:40:51	7301.24,N, 14828.19	08:07:36	7303.29,N, 14835.05

TABLE 15. Transmission schedule, Expt. 8

Time	Transmitting Node	File Time Stamps
22:40 - 23:00	bambi	420 - 479
23:00 - 0:00	dopey	480 - 539
0:00 - 1:00	wally	540 - 599
1:00 - 2:00	no transmitter	600 - 659
2:00 - 3:00	bashful	660 - 719
3:00 - 4:00	doc	720 - 779
4:00 - 5:00	bambi	780 - 839
5:00 - 6:00	no transmitter	840 - 899
6:00 - 7:00	wally	900 - 959

Time	Transmitting Node	File Time Stamps
7:00 - 8:00	gary	960-1019
8:00 - 8:10	bashful	1020-1080

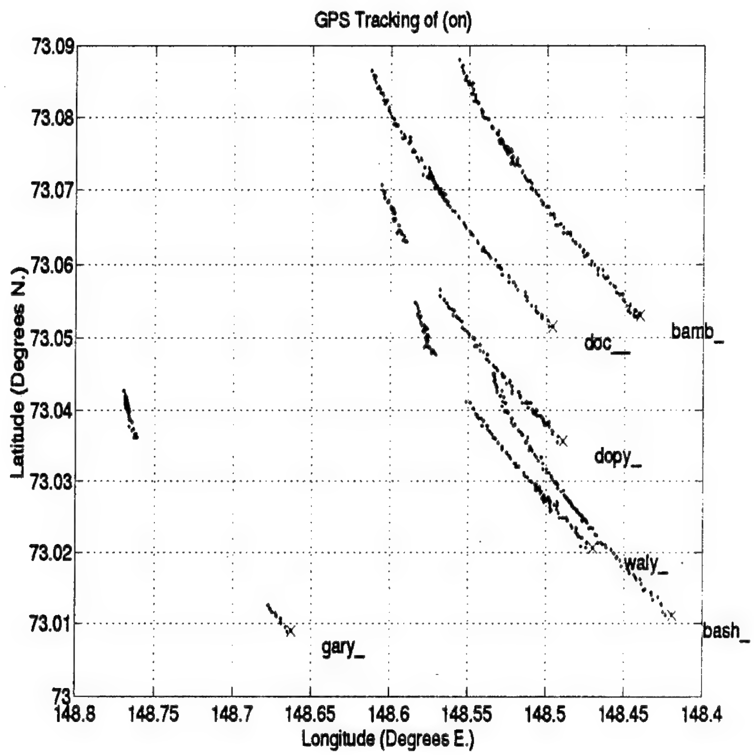

FIGURE 5. GPS tracking, Expt. 8.

6.9 Experiment 9 *(collide)*

Description:	Near-simultaneous transmissions from two modems (dopey and bashful) to remaining modems.
Data purpose:	Analysis of multi-user and full-duplex receiver algorithms.
Started:	10:42, March 30
Duration:	approx. 30 minutes
Number of nodes:	6
Number of channels:	4
Packet length:	2500 symbols
Raw data:	/data0/marky/arctic/collide
Processed data:	/data0/marky/arctic/expt9

TABLE 16. GPS positions, Expt 9.

Node	GPS
bambi	73:05.28N, 148:33.38W
bashful	73:02.70N, 148:32.04W
doc	73:05.19N, 148:36.73W
dopey	73:04.24N, 148:36.33W
gary	73:02.56N, 148:46.16W
wally	73:03.29N, 148:35.05W

7.0 Data Extraction and Multi-channel Receiver Scripts

7.1 MATLAB script for automatic gain control, AGC.M

```
function[y,p] = agc(x, mu)
%
%[y,p] = agc(x, mu)
%Feedforward automatic gain control with parameter mu.
%x is the input signal vector (time series). x may
%be complex.
%mu sets the pole position of the first order input
%power filter - pole is at 1-mu. A small mu
%(e.g., 0.005) results in slow agc action.
%yis the output signal vector
%pis the input signal power estimate.
%
%
%Mark JohnsonAcoustic Telemetry Group, WHOI
%
%Last Modified:24 October 1994
%

% prime power estimate with a power estimate from the
% first ~ 5 time constants of x.

nstart = ceil(5/mu) ;

xs = abs(x).^2 ;
p_start = mean(xs(1:nstart)) ;
start_gain = 1/sqrt(p_start) ;

% compute power estimate for all of x

p = filter([mu 0], [1 -(1-mu)], xs, p_start) ;

% multiply x by inverse sqrt of power estimate to get
% mean power of 1

agco = sqrt(p).^(-1) ;
y = x.*agco ;
```

7.2 MATLAB script for demodulation and decimation, DEMDEC.M

```
function Xdd = demdec(X,fc,fs,df)
%
%Xdd=demdec(X,fc,fs,df)
%Demodulate and decimate a passband signal set.
%
%X is the raw input data sequence in column-signal
%format.
%fc is the carrier frequency.
%fs is the passband sampling frequency.
%df is the decimation factor.
```

```
%
%Xdd is the demodulated and decimated output matrix in
%column-signal format.
%
%
%     Mark Johnson    Acoustic Telemetry Group, WHOI
%
%     Last Modified:  24 October 1994
%
[np nch] = size(X) ;
Xd = X.*(exp(-j*2*pi*fc/fs*(1:np))')*ones(1,nch) ;
Xdd = zeros(np, nch) ;
for i=1:nch,
   Xdd(:,i) = decimate(Xd(:,i),df) ;
end
```

7.3 MATLAB script for the decision feedback equalizer, EQ.M

```
function[W,de,dd,sef,F,fr] = eq(X,T,map,pll,N,options,fcn)
%
%[W,de,dd,se,F,fr]=eq(X,T,map,pll_f,N,options,fcn)
%Multichannel joint fractionally-spaced RLS and PLL equaliser and
%decoder for coherent communications. See Stojanovic et al.
%'', IEEE Trans. ...
%
%X is the complex input data sequence. It has a column
% for each observed signal. The sampling-rate for all
%channels is nnew samples per symbol (see options below).
%The algorithm will attempt to decode Ns symbols where
%Ns = (number of rows in X) / nnew.
%T is the complex training sequence. The initial training
%duration is determined by the size of T.
%map is the complex alphabet vector.
%N is a n+1 vector where n is the number of columns in X.
%The first n elements specify the number of parameters in the
%feedforward filter associated with the corresponding
%column of X. The support of the ith feedforward filter
%is thus N(i)/nnew symbols. The final entry in N specifies the
%number of parameters in the feedback filter. As the decision
%feedback is always at the symbol-rate, the feedback
%support is N(n+1) symbols.
%pll sets the PLL filter coefficients: the first row of pll_f is
%the numerator coefficients. The second row is the denominator.
% Coefficient organisation is as for Matlab's filter.m function.
%options is a vector of options:
%[nnew; lambda; dinit; THRESH; BUDGET; Nrec]
%where (default values are in {}):
%nnew {2} is the number of input measurements to be used
%per symbol. 1/nnew is the fractional spacing.
%lambda {0.995} is the forgetting factor for the RLS algorithm.
%dinit {100}is the initial diagonal value of the inverse
% correlation matrix used in the RLS algorithm.
%THRESH {0}is the mse level in dB above which the equaliser
% parameters are updated using the RLS algorithm
```

```
%after initial training.
%BUDGET {Ns}the maximum number of RLS updates permitted in
%decoding X.
%Nrec {50}sets the period in symbols between parameter value

%reports (see discussion of W below).
%fcn is the name of the .m file to be used for equalizer update

%Available update scripts are 'n2rls', 'n3rls', 'sqrrls'.

%

%W is a matrix each column of which is the RLS parameter vector
% recorded every Nrec symbols.
%de is the Ns vector of equaliser outputs.
%dd is the Ns vector of decision outputs.
%sef is the Ns vector of filtered squared errors.
%F is the Ns x n matrix of the input phase compensation as
%determined by the phase-locked-loop algorithm.
%fris the fraction of decoded symbols for which an RLS update
%was performed. fr <= BUDGET/Ns.
%
%
%     Mark Johnson    Acoustic Telemetry Group, WHOI
%based on rls.m (jointfsrls) by Milica Stojanovic
%
%     Last Modified: 31 October 1994
%


% Initialisation...

sef_num = [0.05 0] ;          % squared-error filter: single pole @ 0.95
sef_denom = [1 -0.95] ;

if nargin<7
fcn = 'n2rls' ;                % default RLS update
end

if nargin<6                   % set default options
nnew = 2 ;
lambda = 0.995 ;
dinit = 1 ;
THRESH = 0 ;
BUDGET = floor(size(X,1)/nnew) ;
Nrec = 50 ;
else
nnew = options(1) ;
lambda = options(2) ;
dinit = options(3) ;
THRESH = options(4) ;
BUDGET = options(5) ;
Nrec = options(6) ;
end
```

```
np = sum(N) ;                  % total number of parameters
Nt = max(size(T)) ;                  % number of training symbols
Ns = floor(size(X,1)/nnew) ;       % total number of symbols to decode
no = size(X,2) ;               % number of feedforward elements (channels)
IN = [0;cumsum(N)] ;
Phi = zeros(np,1) ;     % regressor vector

eval(['[Theta Zeq]=' fcn '(np, dinit);']) ;              % initialize RLS algorithm

update = ['[Theta Zeq]=' fcn '(Phi,e,lambda,Theta,Zeq);'] ;      % script for RLS update

nfill = max(N(1:no)./nnew) ;           % initially hold-off RLS updates for
% nfill symbols until regressor has
% some data in it.
j = sqrt(-1) ;

[f PLLs] = filter(pll(1,:),pll(2,:),0) ;           % initialize PLL filter states and output
PLLs = PLLs*ones(1,no) ;          % ...for each input channel
g = zeros(no,1) ;

de = zeros(Ns,1) ;           % record of equaliser output
sef = zeros(Ns,1) ;          % record of filtered squared estimation errors
dd = zeros(Ns,1) ;            % record of hard decision output
F = zeros(Ns,no) ;            % record of phase-locked loop adjustments
W = zeros(np,floor(Ns/Nrec)) ;          % record of equalizer parameters

[sef(1) Zsef] = filter(sef_num, sef_denom, 0) ;        % initialize squared-error filter states
d = 0 ;                        % 0th feedback symbol is zero
updatecnt = 0 ;               % zero the RLS update counter
k = 1 ;

% Decode Ns symbols...

for i=1:Ns,

% ...compute data symbol estimate and estimation error

for ii=1:no,                 % for each input channel
% ...compute phase-locked loop coefficient

[f PLLs(:,ii)] = filter(pll(1,:),pll(2,:),g(ii),PLLs(:,ii)) ;

F(i,ii) = f ;                 % record PLL coefficient
% ...and update regressor

Phi(IN(ii)+1:IN(ii+1)) = [exp(-j*f)*X((i-1)*nnew+1:i*nnew,ii); Phi(IN(ii)+1:IN(ii+1)-nnew)] ;

% ...apply feedforward filter

p(ii) = Theta(IN(ii)+1:IN(ii+1))'*Phi(IN(ii)+1:IN(ii+1)) ;
end

Phi(IN(no+1)+1:np) = [d ; Phi(IN(no+1)+1:np-1)] ;    % add last symbol as feedback to regressor
```

```
t = Theta(IN(no+1)+1:np)'*Phi(IN(no+1)+1:np) ;     % apply feedback filter
de(i) = sum(p)+t ;                                  % sum partial outputs for symbol estimate

if i>Nt,                                            % determine output symbol
[ee ind] = min(abs(map-de(i))) ;                    % if not training...
d = map(ind) ;                                      % ...find nearest symbol to de in map vector
else
d = T(i) ;                                          % if training pick next training symbol
end

dd(i) = d ;                      % record output symbol
e = d - de(i) ;                  % compute error
% filter squared-error and record

[sef(i) Zsef] = filter(sef_num, sef_denom, e*e', Zsef) ;

% check if RLS update is required

if((i>nfill) & ((updatecnt <= BUDGET) & (sef(i) > THRESH)) | (i<=Nt))

% ...compute RLS parameter update

eval(update) ;                   % do update
updatecnt = updatecnt+1 ;        % ...and increment update count
end

if(rem(i,Nrec)==0)               % report equalizer parameters every Nrec symbols
W(:,k) = Theta ;
fprintf('%d ',i) ;
k = k+1 ;
end

% ...compute PLL update

g = imag(p.*conj(p+e)) ;

end

fprintf('Number of updates = %d\n', updatecnt) ;
fr = updatecnt/Ns ;
```

7.4 MATLAB script to run the equalizer, EQRUN.M

```
% eqrun.m
%Example script for running eq.m receiver and displaying results
%Assumes input variables y (baseband signals) and ps (code detect point)
%obtained using:
%[y ps] = prepinput(x, code, sym_rate) ;
%
%see prepinput.m for further details.
%
%
%       Mark Johnson    Acoustic Telemetry Group, WHOI
%
```

```
%     Last Modified:  31 October 1994
%

sta = ps + nnew*nullsym ;% start point in signal vector

options = [nnew; lambda; dinit; THRESH; BUDGET; 50] ;     % options vector for eq.m

nn = [N(1:size(y,2)).*nnew; N(size(N,1))] ;          % convert feedforward filter symbol supports
                                                     % as given in rx_set to number of parameters

[W de dd sef ph fr] = eq(y(sta:sta+nnew*nsymbols-1,:),train(1:ntrain),map,pll,nn,options,eqfcn);

figure(2);
ne = plotresult(de, dd, sef, ph, train(1:size(dd,1)), ntrain) ;       % plot results in 4-graph format
```

7.5 MATLAB script for locating barker code for synchronization, FIND1STPEAK.M

```
functionps = find1stpeak(dv,thresh)
%
%ps=find1stpeak(dv,thresh)
%Find the first peak in vector dv that exceeds a threshold.
%
%dv is the input 'decision' sequence.
%thresh is the level above which a detection is made.
%
%ps is the index of the first peak in dv.
%
%
%     Mark Johnson    Acoustic Telemetry Group, WHOI
%
%     Last Modified:  25 October 1994
%

% find first point that exceeds the threshold

ps = min(find(dv>=thresh)) ;

% find the immediately following peak

ps = ps + min(find(diff(dv(ps:size(dv,1)))<0)) - 1 ;
```

7.6 MATLAB script for RLS updating of Equalizer parameters, N2RLS.M

```
function[C,R] = n2rls(X,error,lambda,C,R)
%
%[C,R] = n2rls(X,error,lambda,C,R)
%Standard n-squared RLS update implementation
%for real or complex signals.
%X is the regressor vector (nx1)
%erroris the current error value (i.e.,
%desired - C'R)
%lambdais the forgetting factor (0<lambda<1)
%Choose value near 1 (e.g., 0.995) for
```

```
%slow tracking. Tracking time constant
%is approximately 1/(1-lambda) samples.
%Cthe current parameter vector (nx1)
%Rthe current inverse auto-correlation matrix
%estimate (nxn).
%
%[C,R] = n2rls(n, dinit)
%Initializes C with zeros(n,1) and R with diag(dinit)
%
%
%      Mark Johnson    Acoustic Telemetry Group, WHOI
%
%      Last Modified:  24 October 1994
%

if(nargin == 2)
n = X ;
dinit = error ;
R = dinit*eye(n) ;
C = zeros(n,1) ;
else
Z = R*X ;
mu = 1/real(lambda + X'*Z) ;
K = mu*Z ;
R = (R - K*Z')/lambda ;
C = C + error'*K ;
end
```

7.7 MATLAB script for plotting the input, PLOTINPUT.M

```
functionplotinput(x,y,p,ire,ps,settings)
%
%plotinput(x,y,p,ire,ps,settings)
%display input signal properties for coherent receiver.
%To be used with prepinput.m. See prepinput.m for details.
%xis the complex input data sequence. It has a column
% for each observed signal.
%yprocessed signal
%pestimated input signal power from agc.m
%psstart point in x found using find1stpeak.m
%irechannel impulse response estimate from cre.m
%settings settings vector (see prepinput.m)
%
%
%      Mark Johnson    Acoustic Telemetry Group, WHOI
%
%      Last Modified:  31 October 1994
%

symrate = settings(1) ;
intf = settings(2) ;
ire_display = settings(6) ;

clg
```

```
ns = size(x,1) ;
ind = (0:ns-1)/(symrate*intf) ;
subplot(221), plot(ind,abs(x(:,1))), grid, title('raw time signal')
xlabel('seconds'), ylabel('amplitude')

subplot(222), plot(ind,abs(y(:,1))), grid, title('processed time signal')
xlabel('seconds'), ylabel('amplitude')

subplot(223), plot(ind,p), grid, title('Input power estimate')
xlabel('seconds'), ylabel('power')

nire = ceil(symrate*intf*ire_display) ;
subplot(224), plot(((0:nire-1)-ps)/(symrate*intf), ire(1:nire)), grid
title('Channel impulse response estimate')
xlabel('time in seconds'), ylabel('amplitude')
```

7.8 MATLAB script for plotting results, PLOTRESULT.M

```
function ne = plotresult(de,dd,sef,ph,train,ntrain)
%
%ne = plotresult(de,dd,sef,ph,train,ntrain)
%display results of coherent receiver, eq.m
%de     is the vector of equaliser outputs.
%    dd    is the vector of decision outputs.
%    sef   is the vector of filtered squared errors.
%    ph    is the matrix of the input phase compensation coefficients.
%trainis the vector of transmitted symbols
%ntrainis the number of symbols to be used for training
%
%neis the number of symbol errors in the decoded packet
%
%
%    Mark Johnson    Acoustic Telemetry Group, WHOI
%
%    Last Modified:  31 October 1994
%


nsymbols = size(de,1) ;
ne = [] ;

clg
subplot(221), plot(10*log10(sef)), grid, title('MSE')
xlabel('time in symbols'), ylabel('dB')

subplot(222), plot(de(ntrain+1:nsymbols),'g+'), grid
axis([-2 2 -2 2]), axis(axis), title('Constellation')
xlabel('real(de)'), ylabel('imag(de)')

subplot(223), plot(ph), grid, title('Phase')
xlabel('time in symbols'), ylabel('radians')

if nargin == 6
errors = abs(dd - train) >= 1 ;
```

```
ne = sum(errors(ntrain+1:nsymbols)) ;
fprintf('number of symbol errors after training = %d\n', ne) ;
subplot(224), plot(errors), grid, title('Error distribution')
xlabel('time in symbols'), ylabel('errors')
end

drawnow;
```

7.9 MATLAB script for preparing the input file, PREPINPUT.M

```
function [y,ps,p] = prepinput(x,code,settings)
%
%[y,ps,p] = prepinput(x,code,settings)
%Prepare input signal to equalizer by (i) hardlimiting,
%(ii) start detection, and (iii) agc
%xis the complex input data sequence. It has a column
% for each observed signal. The sampling-rate for all
%channels is nnew samples per symbol (see settings below).
%codethe complex start code at a sampling-rate of intf (see
%below) input samples per code symbol.
%settings settings vector containing:
%settings = [symrate intf limit start_thresh agc_mu ire_display]
%where:
%symrateis the code symbol-rate in Hz
%intfis the number of input samples per code symbol
%limitthe hard limit threshold as a percentage of input
%events.
%start_threshthreshold above which a start may be detected.
%See find1stpeak.m for details.
%agc_mucontrols pole position of agc power filter. See
%agc.m for details.
%ire_displayhow many seconds of the impule response estimate
%to display.
%
%yprocessed signal
%psstart point in y according to cre.m and find1stpeak.m. ps is in
%input samples not symbols.
%pestimated input signal power from agc.m
%
%
%     Mark Johnson    Acoustic Telemetry Group, WHOI
%
%     Last Modified:  31 October 1994
%

nsearch = 1000 ;

intf = settings(2) ;
limit = settings(3) ;
start_thresh = settings(4) ;
agc_mu = settings(5) ;

xl = rmoutliers(x,limit) ;% hard limit
codedf = interp(code,intf) ;% interpolate start code
```

```
ire = cre(xl(1:nsearch), codedf, 1) ;% cross-correlate code and input

[y p] = agc(xl, agc_mu) ;% agc input

dv = ire./p(1:nsearch) ;% decision vector = cross-correlator
% output / input power
ps = find1stpeak(dv, start_thresh) ;% find start peak in decicion vector

if ps ~= []
fprintf('start found at input point %d\n', ps) ;
else
fprintf('no start found with threshold = %d\n', start_thresh) ;
end

figure(1) ;% ...plot results

plotinput(x, y, p, ire, ps, settings) ;
```

7.10 MATLAB script for converting raw data to complex vector, RAW2VEC.M

```
function x=raw2vec(v)
%
%x=raw2vec(v)
%Convert 2-column format raw data from AMS V1.0 receiver
%into multi-channel signal matrix. Supports files from
%Lake Winnepesaukee (Feb. 1994) and Arctic (March 1994)
%expeditions.
%
%vis the raw data (a 2-column real matrix)
%xis the complex recovered time series in column-
%channel format
%
%
%      Mark Johnson    Acoustic Telemetry Group, WHOI
%
%      Last Modified:  24 October 1994
%

RXM_MAGIC = 1234 ;
BUFSPACE = 100 ;
x = [] ;

if(v(1,1) == RXM_MAGIC)

% Arctic vintage data with circular buffer structure
% header contains:
%
%magic numbertime (secs)
%data enddata start
%ch1 detectch1 time offset
%   :    :
%chn detectchn time offset
%
% remainder of file is [real imag] per row
```

```
nch = v(3,1) ;
pos = v(2,:) + 1 ;
n = size(v,1) ;
st = nch + 4 ;
y = v(st:n,1) + j*v(st:n,2) ;
n = size(y,1) ;
fprintf('Arctic data file: %d channels, %d samples\n', nch, n/nch) ;

first = pos(2)-BUFSPACE*nch ;
if(first < 1)
first = n+first ;
end

if(pos(1) == 1)
y = y(first:n) ;
else
y = y([first:n 1:pos(1)-1]) ;
end

for i=1:nch,
x = [x y(i:nch:size(y,1))] ;
end

else
% raw data is Winnepesaukee vintage

fprintf('Pre-Arctic single-channel data file\n') ;
pos = v(1,:)+1 ;
n = size(v,1) ;
x = v(2:n,1) + j*v(2:n,2) ;
if(pos(1) == 1)
x = x(pos(1):n-1) ;
else
x = x([pos(1):n-1 1:pos(1)-1]) ;
end
end
```

7.11 MATLAB script for removing outliers from data set, RMOUTLIERS.M

```
function y = rmoutliers(x, thresh)
%
%y = rmoutliers(x, thresh)
%Remove outliers from time series, x, using symmetric
%hard limiter with limit thresholds set from the
%histogram of x.
%
%x is the signal in column vector format (maybe
%real or complex)
%thresh is the clipping threshold in percentage of
%input signal events, e.g., thresh = 0.01 sets
%the limiting level such that only 0.01

% of
```

```
%input points will be clipped.
%
%y is the output column vector.
%
%
%     Mark Johnson    Acoustic Telemetry Group, WHOI
%
%     Last Modified: 24 October 1994
%

nbins = 50 ;% resolution of histogram

ax = abs(x) ;
[n, xdist] = hist(ax, nbins);
lt = xdist(max(find(100*(1-cumsum(n)/sum(n)) > thresh))) ;
y = (ax <= lt).*x + lt*(ax > lt).*sign(x) ;
```

7.12 MATLAB script with example settings for single channel reception, TEST_SET.M

```
%rx_set.m
%standard receiver settings for eq.m and prepinput.m
%
%
%     Mark Johnson    Acoustic Telemetry Group, WHOI
%
%     Last Modified: 31 October 1994
%


load train1          % training symbols

% settings for prepinput.m

symrate = 2500 ;          % symbol rate (Hz)
limit = 0.04 ;        % clipping threshold in % of events
agc_mu = 0.005 ;        % sets pole position of agc power filter
% - pole is at 1-agc_mu
start_thresh = 100 ;          % power ratio threshold for detector
ire_display = 0.02 ;          % maximum length of impulse response to display (secs)
intf = 2 ;          % number of samples per code symbol in input
% ...settings vector for prepinput.m

code = [-1;1;-1;1;-1;-1;1;1;-1;-1;-1;-1;-1] ;          % 13-element barker start code

settings = [symrate intf limit start_thresh agc_mu ire_display] ;


% settings for eqrun.m

N = [5;9] ;          % equalizer feedforward and feedback support
% in symbols.
map = [1+j;-1-j;-1+j;1-j];          % map vector for QPSK
pll = [0.011 -0.01 0 ; 1 -2 1] ;          % pll filter numerator and denominator
```

```
% ...this pll has two integrators and one zero.
nullsym = 20 ;          % number of symbols between start code and data
THRESH = 10^(-7.5/10) ;          % mse threshold in dB - equalizer parameters are only
% adjusted when the current mse is above THRESH
nsymbols = 1000 ;          % number of symbols to decode
nnew = 2 ;          % number of samples per symbol in input
BUDGET = nsymbols ;          % maximum number of RLS updates to perform
ntrain = 200 ;          % number of training symbols at start of packet
lambda = 0.995 ;          % RLS forgetting factor
dinit = 1000 ;          % RLS initial regressor power value
eqfcn = 'n2rls' ;          % RLS update function to use.
```

7.13 Source Code for extracting: Time, Latitude and Longitude from STATUS.LOG files:

```
/*-----------------------------------------------------------------------------------------------*/
/*STATFILT.C - This code extracts time, latitude and longitude from          */
/* STATUS.LOG files created during the Arctic deployment of March 1994          */
/* The program removes any measurements which were taken with a TFOM > 10  */
/* See Magellan data sheet for details on TFOM          */
/*          */
/* Author: D. Herold, WHOI          */
/* Date: 4/26/94          */
/*-----------------------------------------------------------------------------------------------*/

#include <stdio.h>
#include <ctype.h>
#include <stdlib.h>

#define CORRECT_READS 3

main(argc,argv)

int argc;
char *argv[];

{
FILE *fname1;          /* file pointers*/
FILE*fname2;
char in_str[80];          /* input string*/
inti;          /* declare vars.*/
inttfom;
inthour,minute,second;
floatlatitude,longitude;

if ( argc != 3 )          /* check for correct usage*/
        {
        fprintf(stdout,"Usage: statfilt <file1> <file2>\n");
        fprintf(stdout,"NOTE: <file1> must be a dos2unix'd status.log file....\n");
        exit(0);
        }

/* open files*/
```

```
   if ( (fname1 = fopen(argv[1],"r")) == NULL )
          {
          exit(0);
          }
if ( (fname2 = fopen(argv[2],"w")) == NULL )
          {
          exit(0);
          }

/* read file and get pertinent information, store in file2*/

while ( fscanf(fname1,"%s",in_str) != EOF )
{

if (CORRECT_READS == sscanf(in_str,"%2d:%2d:%2d ",&hour,&minute,&second) )
  {
  for ( i = 0; i <= 2; i++)
  fscanf(fname1,"%s",in_str);
  sscanf(in_str,"%x",&tfom);
  if (tfom < 10)
    {
    fscanf(fname1,"%s",in_str);
    sscanf(in_str,"%f,N,%f",&latitude,&longitude);
    fprintf(fname2,"%2d %2d %2d  ",hour,minute,second);
    fprintf(fname2,"%7.2f %8.2f\n",latitude,longitude);
    }
  }

}

fclose(fname1);/* close files*/
fclose(fname2);

}          /* exit*/
```

7.14 MATLAB Script for for correcting and plotting GPS Lattitude and Longitudes extracted from STATUS.LOG files.

```
% gpsplot.m     D. Herold, WHOI     26 April 1994
% This file loads data from an Arctic status.log file which
% has been dos2unixed and then run though the statfilt.exe
% program in ~/aalan/arctic/gpsdata/  directory.
%
% function [pos,samples,etime] = gpsplot(experiment)
%
function [pos,samples,etime] = gpsplot(experiment)

nodes = ['bamb_'
 'bash_'
 'doc__'
 'dopy_'
 'gary_'
```

```
'waly_'] ;

nodenum = 0 ;
x_min = 999 ;
x_max = -999 ;
y_min = 999 ;
y_max = -999 ;

pos = zeros(150,12) ;
etime = zeros(150,6) ;

clg

for k = 1:length(nodes)
  file = [nodes(k,:) experiment '_gps'] ;
  filename = [file '.log'] ;
  if exist(filename)
    filename
    nodenum = nodenum + 1 ;
    load(filename) ;
    eval(['y =' file ';' ]) ;

    % get the elapsed experiment time

    tm = y(:,1).*60+y(:,2)+y(:,3)./60 ;
     tm(find(tm==max(tm))+1:length(tm)) = tm(find(tm==max(tm))+1:length(tm))
    + 24*60 ;
    etime(1:length(tm),nodenum) = tm ;

   % Convert to decimal degrees

    y(:,4) = fix(y(:,4)./100)+(rem(y(:,4)./100,1)./60).*100 ;
    pos(1:length(y),2*nodenum-1) = y(:,4) ;
    y(:,5) = fix(y(:,5)./100)+(rem(y(:,5)./100,1)./60).*100 ;
    pos(1:length(y),2*nodenum) = y(:,5) ;
    samples(nodenum) = length(y) ;
    index(nodenum) = k;

    % Get max and min for axes

     x_min = min(x_min,min(pos(1:samples(nodenum),2*nodenum))) ;
     x_max = max(x_max,max(pos(1:samples(nodenum),2*nodenum))) ;
     y_min = min(y_min,min(pos(1:samples(nodenum),2*nodenum - 1))) ;
     y_max = max(y_max,max(pos(1:samples(nodenum),2*nodenum - 1))) ;
  else
     k = k + 1 ;
  end

end

x_min = x_min - 0.01 ;
x_max = x_max + 0.01 ;
y_min = y_min - 0.01 ;
y_max = y_max + 0.01 ;
```

```
delta_lat = y_max-y_min ;
x_max = x_min + delta_lat / cos(y_max*(pi/180)) ;

figure(1) ;   % plot out gps positions of nodes

axis('square');
axis('equal');
axis([x_min x_max y_min y_max]);

for k = 1:nodenum
  plot(pos(1:samples(k),2*k),pos(1:samples(k),2*k-1),'.'), grid on
  hold on
  plot(pos(1,2*k),pos(1,2*k-1),'x')
  text(pos(1,2*k)-0.02,pos(1,2*k-1)-0.002,nodes(index(k),:))
end

view(180,-90)

Title = ['GPS Tracking of (' experiment ')'] ;

title(Title)
xlabel('Longitude (Degrees E.)')
ylabel('Latitude (Degrees N.)')
hold off

figure(2)   % not required, however, lets look at times of experiments

for k = 1:nodenum
  plot(etime(1:samples(k),k)-max(min(etime)),'.'), grid on
  hold on
  text(samples(k),etime(samples(k),k)-max(min(etime)),nodes(index(k),:))
end

Title = ['GPS Time vs. Sample of (' experiment ')'] ;

title(Title)
xlabel('Sample')
ylabel('Sample Time (Minutes)')
hold off

figure(1)
```

Appendix A - Log Book Notes

The following information is extracted from the log book kept during the experiment.

0.1 March 17, 1994

- 1700 MJ, DH Arrive Deadhorse, AK.
- 1900 Move equipment into lab space.

0.2 March 18, 1994

- 0730 Move all equipment into new lab space and unpack.
- 0900 Hook up TELEM (SPARC 10), restore new data from backups of work done at WHOI after air shipment. Having the same problem with the DOS shell program. What you type in the dos window is not displayed on the screen. Called BE at WHOI. There was no easy way to fix the problem. Decided to use JOSKO (SPARC IPC) as we had in Winnipesaukee experiment.
- 1130 Lunch
- 1200 Finished switching IPC in for SPARC 10
- 1500 Sneezy, Wally, Doc all Tx and Rx ok. Hook up GPS to Sneezy to see if we get reception inside builing.
- 1615 Start bringing in PVC tubes for thru-ice fittings to warm them up prior to assembly.

0.3 March 19, 1994

- 0800 No journal entries for this day. Presume we spent it chasing apparent software bug.

0.4 March 20, 1994

- 1700 JC arrives Deadhorse
- 2000 Determine that problems which have looked like transmitter and receiver software problems are actually hardware problems not previously encountered on A/D board. We have to date been unlucky and only tried combinations of modems with bad boards.

0.5 March 21, 1994

- 0200 Finish debugging and preparing six of the modems for deployment initially on ice. Pack equipment for trip to ice.
- 0900 Depart Deadhorse for Ice Camp East.
- 1100 Arrive Ice camp, unload plane, move equipment to our ice hut.
- 1300 Short lunch break

- 1400 Raise base-camp antenna, organize hut, get base camp electronics running. We are having problems with the UPS, we keep getting "power hits". We assume that it is due to us overloading the UPS.

0.6 March 22, 1994

- 0800 It turns out that the camp generator does not have a closely regulated frequency and it is too difficult for the UPS operate correctly.
- 0830 Deployed Doc ~50m from camp: GPS position 7257.16,N,14822.90 Must prepare pvc tube endcaps inside after they have warmed up inside the huts. Do NOT use teflon pipe sealer.
- 1300 Deploy Bambi, GPS 7258.04,N,14827.70

0.7 March 23, 1994

- 1300 Deploy Bashful, GPS 7256.58,N,14823.83

0.8 March 24, 1994

0.9 March 25, 1994

- 0900 DH departs Ice camp. ED arrives.

0.10 March 26, 1994

0.11 March 27, 1994

- 0930 Did temperature log. Bambi taken by helicopter to replace gary who wend down yesterday. onite batch file running on dopey, gary and doc, failed after first iteration. Made some changes to hrt.bat (overnite batch file) adn renamed collect.bat. This needs the following environment variables: TXSEQ (>=0., < NNODES). NNODES e.g. 4. PRESLEEP (sleep time between Rx and copy). RXSLEEP (sleep time in seconds between Rx's). TXSLEEP (sleep time in seconds beteen Tx's. DATADIR (destination data directory on e:. These are all set in onset.bat which is called from autorun.bat. Also in mailbox.bat @ start there is a checd for e:%sysdir/collect.mbx, only if it exists can you call collect.
- 1000 Gary replaced by Bambi. Bambi came up good. New GPS fix: 7259.65,N,14814.10. Temperatures: 17.4, 25.5 -14.3, -22.7
- 1030 File set required for collect is: collect.bat, onset.bat, autorun.bat, mailbox.bat, lsleep.bat, remain.bat, rx_mess.bat
- 1030 Checked LSLEEP with env. variables, works ok.
- 1215 Foung problem with gary was (suspected) faulty ARLAN. Repalced with ARLAN from Wally. Also, deadperson battery was flat, replaced and checked thru checklist ok. Took gary to new site with helicopter. Able to ARLAN communicate ok. GPS is 7257.10,N,14824.42

- 1403 MIT1 data set taken?
- 1723 MIT1 data set done?
- 2236 Temperature log
- 2240 Run COLLECT

0.12 March 28, 1994

- 0845 Stopped COLLECT - all except Doc returned. Also doc did not collect any paketes. All data and logs are in : %datadir%/on27_3.

TABLE 17. Transmitters for 2240 COLLECT

GPS Hour	Transmitter	Time Stamp
7	bambi	420-479
8	bashful	480-539
9	gary	540-599
10		600-659
11	dopey	660-719
12	bambi	720-779
13	bashful	780-839
14	gary	840-899
15		900-959
16	dopey	960-1019
17	bambi	1020-1079

- 1446 Datasonics modem on Odyssey is doing occasional broadcasts for next two hours - may be received by the long packet capture on gary, dopey, and bashful (on28_3)
- 1500 Field service on bambi (copied new auitorun.bat) installed doc in new site.
- 1500 Installed wally in camp site (was doc site)
- 1610 GPS log
- 1700 Temperature log
- 2100 Temperature log
- 2130 GPS log

• 2200 GPS get_xyz.

TABLE 18. XYZ Positions

NODE	X	Y	Z
doc	-1588321.42	-991950.08	6076456.88
bambi	-1587216.75	-993415.57	6076521.02
dopey	-1589628.87	-993049.73	6075957.36
wally	-1590715.19	-994423.98	6075441.62
gary	-1595155.84	-989769.79	6075038.56
bashful	-1590679.12	-996339.06	6075141.54

• 2200 captured 2 data sets from modem on odessey. These are 4 channel 1K symbols and are stored in MIT2 directories.

• 2222 COLLECT started with 4 channel rxd1.m

0.13 March 29, 1994

• 0830 Stopped COLLECT (doc and gary had reset earlier ~1540 rather than 1730).

• 0830 Temperature log:

• 0830 GPS positions:

• 0830 GPS get_xyz:

• 0930 Re-start COLLECT with multi-channel short packets. Last nights collect results are stored in ON28_3_A directories. Transmit schedule was:

TABLE 19. Transmitter schedule

GPS Hour	Transmitter	OUT Keys
7	bambi	420:479
8	dopey	480:539
9	wally	540:559
10	gary	600:659
11	bashful	660:719
12	doc	720:779
13	bambi	780:839
14	dopey	840:899
15	wally	900:959
16	gary	960:1019
17	bashful	1020:1079

- 1705 Stopped data capture. All units returned.
- 1705 Temperature log.
- 1711 GPS Positions.
- 1717 Re-started COLLECT, (short packet, 4 channel) with reduced PRESLEEP, RXSLEEP increments (20 of 25 previously) and reduced TXSLEEP (40 of 50 previously).
- 1845 MIT doing Odyssey ops with datasonics modem on (1 per minute).
- 1915 MIT stops Odyssey.
- 2130 COLLECT stopped (multichannel, short packets) stored in ON29_3 directory.
- 2130 GPS log
- 2130 Temperature log
- 2245 COLLECT started with single channel long packets. Previous collect stored in ON29_3 directory.

0.14 March 30, 1994

- 1005 Flow is cracking. ED recovering nodes electronics. Last nites collect is in ON directories (status.log's are still on PCMCIA) Did some collection for milica: dopey and bashful transmitting, remainder receiving. Data is 4 channel, 3000 symbols (2500 transmitted), stored in root directories. e.g. gary_d.
- 1015 TARd directories to tapes.

Appendix B - Wind Speeds

The wind measurement station was by the helo hut until about 5 or 6 on 29 March, so the early wind directions aren't too good but should be in the right ballpark. Times are GMT. The station should have taken one sample per hour, but its timer ran roughly a minute/hour slow(sometimes worse). The 10-minute sample interval was controlled by a more accurate clock, so that windspeed calculations should be OK. Wind direction is TO, not from, kind of true (magnetic+30 degrees). Floe rotation is already included in wind direction.

YrMoDaHrMi	AirT(C)	WSpd(m/s)	WDir	Comp(Floe Rotation)
9403270515	-27.36	1.76	280.00	22.40
9403270618	-26.40	3.04	274.40	21.00
9403270720	-25.86	2.40	271.60	21.00
9403270821	-23.34	0.80	273.00	21.00
9403270923	-25.38	2.40	259.00	21.00
9403271025	-27.96	0.96	235.20	22.40
9403271127	-28.20	2.88	27.80	22.40
9403271231	-29.76	1.12	50.20	22.40
9403271331	-30.54	1.44	257.60	22.40
9403271432	-30.06	2.24	329.00	22.40
9403271535	-30.42	2.88	267.40	22.40
9403271739	-29.22	7.04	18.00	19.60
9403271841	-28.14	6.88	260.40	21.00
9403271943	-27.42	7.84	257.60	19.60
9403272044	-26.82	7.20	266.00	18.20
9403272148	-26.52	7.52	266.00	18.20
9403272248	-26.34	8.96	266.00	19.60
9403272350	-26.52	8.32	270.20	18.20
9403280054	-26.94	11.52	268.80	18.20
9403280154	-27.12	6.56	270.20	18.20
9403280256	-27.96	3.52	274.40	19.60
9403280358	-28.98	3.68	274.40	19.60
9403280459	-29.58	5.28	273.00	19.60
9403280502	-29.58	5.28	273.00	19.60
9403280602	-30.00	5.28	275.80	21.00
9403280704	-30.30	2.72	282.80	21.00
9403280809	-30.60	2.24	278.60	21.00
9403280909	-30.84	2.24	281.40	21.00
9403281009	-31.14	2.24	292.60	21.00
9403281113	-31.38	2.40	302.40	21.00
9403281213	-31.74	1.60	281.40	21.00
9403281315	-31.98	1.44	294.00	21.00
9403281417	-32.40	1.60	277.20	19.60
9403281519	-32.04	1.12	278.60	21.00
9403281621	-31.20	0.96	277.20	18.20
9403281723	-31.20	0.96	277.20	18.20
9403281825	-30.30	1.12	303.80	19.60

YrMoDaHrMi	AirT(C)	WSpd(m/s)	WDir	Comp(Floe Rotation)
9403281926	-29.52	0.48	284.20	18.20
9403282028	-28.20	0.96	305.20	16.80
9403282130	-28.26	2.56	259.00	16.80
9403282232	-27.96	2.56	260.40	16.80
9403282334	-27.30	1.92	256.20	16.80
9403290047	-20.52	15.68	259.00	16.80
9403290552	-26.40	2.88	287.00	58.80
9403290654	-25.74	3.52	294.00	58.80
9403290755	-24.96	4.32	294.00	58.80
9403290859	-24.18	4.64	287.00	58.80
9403290959	-23.10	5.12	282.80	58.80
9403291000	-23.10	5.12	282.80	58.80
9403291101	-22.14	6.56	289.80	57.40
9403291203	-21.42	8.48	289.80	57.40
9403291305	-21.60	8.96	292.60	57.40
9403291409	-22.20	10.72	289.80	57.40
9403291509	-21.42	10.56	288.40	57.40
9403291610	-20.88	17.44	289.80	57.40
9403291713	-20.10	16.16	292.60	57.40
9403291814	-19.20	15.52	292.60	57.40
9403291917	-18.48	15.04	288.40	57.40
9403292017	-17.88	15.84	292.60	57.40
9403292120	-17.34	18.72	296.80	57.40
9403292221	-16.56	20.96	296.80	57.40
9403292323	-15.96	19.04	294.00	57.40
9403300027	-15.78	21.76	299.60	57.40
9403300127	-15.30	18.56	296.80	57.40
9403300228	-15.06	15.36	302.40	57.40
9403300331	-14.82	17.28	302.40	57.40
9403300431	-14.70	20.16	305.20	57.40
9403300534	-15.84	16.80	309.40	58.80
9403300635	-15.84	16.80	309.40	58.80
9403300738	-16.56	10.40	309.40	58.80
9403300841	-17.16	11.68	305.20	58.80
9403300941	-17.40	11.84	309.40	58.80
9403301045	-17.52	9.60	309.40	58.80
9403301145	-17.70	14.24	312.20	58.80
9403301246	-17.94	7.84	312.20	58.80
9403301349	-17.82	6.56	310.80	58.80
9403301449	-17.64	7.04	317.80	58.80
9403301554	-17.16	7.04	319.20	58.80
9403301654	-17.04	5.28	324.80	58.80
9403301755	-16.50	4.32	330.40	57.40
9403301859	-16.20	3.52	350.00	57.40
9403301959	-15.42	2.40	6.80	58.80
9403302001	-15.42	2.40	6.80	58.80
9403302101	-15.00	1.60	313.60	57.40

YrMoDaHrMi	AirT(C)	WSpd(m/s)	WDir	Comp(Floe Rotation)
9403302203	-14.88	2.08	340.20	58.80
9403302304	-14.22	2.72	62.80	58.80
9403310008	-14.58	2.88	58.60	58.80
9403310108	-14.46	3.20	48.80	58.80
9403310209	-14.40	4.16	296.80	58.80
9403310313	-14.34	4.64	299.60	58.80
9403310413	-16.08	6.24	310.80	58.80
9403310517	-17.52	5.92	309.40	58.80
9403310617	-18.06	6.08	310.80	60.20
9403310719	-19.56	5.44	315.00	60.20
9403310822	-19.50	7.20	310.80	60.20
9403310922	-20.34	7.20	316.40	60.20
9403311025	-21.84	5.60	315.00	60.20
9403311125	-22.68	4.80	309.40	60.20
9403311228	-23.46	3.68	299.60	60.20
9403311331	-23.82	4.32	310.80	60.20
9403311431	-24.48	3.20	305.20	60.20
9403311534	-25.26	2.56	302.40	60.20
9403311634	-24.96	2.72	299.60	60.20
9403311736	-24.72	2.56	303.80	58.80
9403311841	-23.88	3.20	310.80	58.80
9403311941	-23.52	2.88	303.80	58.80
9403312044	-22.98	3.04	295.40	58.80
9403312144	-22.44	2.24	298.20	57.40
9403312246	-21.84	2.40	298.20	57.40
9403312349	-21.42	2.24	305.20	57.40
9404010049	-20.88	1.92	315.00	57.40
9404010151	-20.70	1.44	201.40	57.40
9404010254	-21.00	1.28	180.40	57.40
9404010354	-21.60	0.16	207.00	58.80
9404010458	-22.56	0.00	313.60	58.80
9404010558	-22.56	0.48	268.80	58.80
9404010600	-22.56	0.48	268.80	58.80
9404010700	-23.70	0.80	268.80	60.20
9404010804	-24.72	0.80	256.20	60.20
9404010904	-24.00	1.28	285.60	60.20
9404011008	-24.72	2.08	312.20	60.20
9404011108	-24.30	1.76	289.80	60.20
9404011209	-25.38	1.44	278.60	58.80
9404011313	-24.84	1.76	288.40	58.80
9404011413	-24.90	2.56	292.60	60.20
9404011517	-25.08	2.24	303.80	60.20
9404011617	-24.54	2.24	301.00	60.20
9404011719	-24.00	2.72	287.00	58.80
9404011822	-23.10	3.20	288.40	58.80
9404011922	-22.14	4.00	295.40	58.80
9404012026	-20.94	3.84	292.60	58.80

YrMoDaHrMi	AirT(C)	WSpd(m/s)	WDir	Comp(Floe Rotation)
9404012126	-19.80	4.96	291.20	58.80
9404012228	-19.26	5.12	282.80	58.80
9404012331	-19.02	5.76	277.20	58.80
9404020031	-18.66	6.56	291.20	58.80
9404020134	-18.60	6.56	284.20	58.80
9404020237	-18.78	7.04	292.60	58.80
9404020337	-19.26	7.20	289.80	58.80
9404020441	-20.04	7.20	284.20	58.80
9404020541	-20.10	8.32	287.00	58.80
9404020642	-20.10	8.80	289.80	58.80
9404020745	-19.92	7.68	292.60	58.80
9404020845	-19.20	9.44	298.20	57.40
9404020949	-19.38	12.96	291.20	57.40
9404021049	-19.86	12.80	289.80	58.80
9404021152	-19.02	11.04	294.00	58.80
9404021253	-18.18	19.84	295.40	57.40
9404021355	-18.24	20.32	298.20	57.40
9404021458	-17.58	20.48	320.60	57.40
9404021558	-17.34	20.64	299.60	57.40
9404021600	-17.34	20.64	299.60	57.40
9404021700	-16.92	20.16	299.60	56.00
9404021804	-16.38	20.48	323.40	56.00
9404021904	-15.48	21.92	299.60	57.40
9404022006	-14.70	21.92	298.20	57.40
9404022107	-13.92	23.68	298.20	56.00
9404022209	-13.56	25.28	302.40	57.40
9404022312	-13.32	22.56	326.20	57.40

Acknowledgments

Funding for this project was provided by the Office of Naval Research under Contract No. N00014-93-10988 and the Advanced Research Projects Agency under Contracts MDA972-91-J-1004 and MDA972-93-10019.

DOCUMENT LIBRARY

Distribution List for Technical Report Exchange – May 1995

University of California, San Diego
SIO Library 0175C
9500 Gilman Drive
La Jolla, CA 92093-0175

Hancock Library of Biology & Oceanography
Alan Hancock Laboratory
University of Southern California
University Park
Los Angeles, CA 90089-0371

Gifts & Exchanges
Library
Bedford Institute of Oceanography
P.O. Box 1006
Dartmouth, NS, B2Y 4A2, CANADA

Commander
International Ice Patrol
1082 Shennecossett Road
Groton, CT 06340-6095

NOAA/EDIS Miami Library Center
4301 Rickenbacker Causeway
Miami, FL 33149

Research Library
U.S. Army Corps of Engineers
Waterways Experiment Station
3909 Halls Ferry Road
Vicksburg, MS 39180-6199

Institute of Geophysics
University of Hawaii
Library Room 252
2525 Correa Road
Honolulu, HI 96822

Marine Resources Information Center
Building E38-320
MIT
Cambridge, MA 02139

Library
Lamont-Doherty Geological Observatory
Columbia University
Palisades, NY z10964

Library
Serials Department
Oregon State University
Corvallis, OR 97331

Pell Marine Science Library
University of Rhode Island
Narragansett Bay Campus
Narragansett, RI 02882

Working Collection
Texas A&M University
Dept. of Oceanography
College Station, TX 77843

Fisheries-Oceanography Library
151 Oceanography Teaching Bldg.
University of Washington
Seattle, WA 98195

Library
R.S.M.A.S.
University of Miami
4600 Rickenbacker Causeway
Miami, FL 33149

Maury Oceanographic Library
Naval Oceanographic Office
Building 1003 South
1002 Balch Blvd.
Stennis Space Center, MS, 39522-5001

Library
Institute of Ocean Sciences
P.O. Box 6000
Sidney, B.C. V8L 4B2
CANADA

Library
Institute of Oceanographic Sciences
Deacon Laboratory
Wormley, Godalming
Surrey GU8 5UB
UNITED KINGDOM

The Librarian
CSIRO Marine Laboratories
G.P.O. Box 1538
Hobart, Tasmania
AUSTRALIA 7001

Library
Proudman Oceanographic Laboratory
Bidston Observatory
Birkenhead
Merseyside L43 7 RA
UNITED KINGDOM

IFREMER
Centre de Brest
Service Documentation - Publications
BP 70 29280 PLOUZANE
FRANCE

50272-101

REPORT DOCUMENTATION PAGE	1. REPORT NO. WHOI-95-04	2.	3. Recipient's Accession No.

4. Title and Subtitle Report on the Acoustic Network Arctic Deployment	5. Report Date March, 1995
	6.
7. Author(s) Mark Johnson, David Herold, Josko Catipovic	8. Performing Organization Rept. No. WHOI-95-04
9. Performing Organization Name and Address Woods Hole Oceanographic Institution Woods Hole, Massachusetts 02543	10. Project/Task/Work Unit No.
	11. Contract(C) or Grant(G) No. (C) N00014-93-1-0988 (G) MDA972-91-J-1004 MDA972-93-1-0019
12. Sponsoring Organization Name and Address Office of Naval Research and the Advanced Research Projects Agency	13. Type of Report & Period Covered Technical Report
	14.

15. Supplementary Notes

This report should be cited as: Woods Hole Oceanog. Inst. Tech. Rept., WHOI-95-04.

16. Abstract (Limit: 200 words)

This report describes the March 1994 Arctic deployment undertaken by the Acoustic Telemetry Group of WHOI. The deployment was a part of the 1994 Sea Ice Mechanics Initiative (SIMI) project and was based at the west SIMI camp, approximately 150 nautical miles north-east of Prudhoe Bay, Alaska. The goal of the deployment was to install a network of six high-performance acoustic modems, developed at WHOI, and to obtain a data set demonstrating the communications and acoustic monitoring capabilities of the network.

The six modems in the network were deployed over an area of 22 square km and communicated via radio Ethernet with a computer at the SIMI camp. Each model had a global positioning system, an acoustic source and an 8 element receiving array. The network was operated in a round-robin broadcast mode (i.e., each modem in turn transmitted a packet of data while the others received). The transmissions were 5000 bits-per-second QPSK with a 15kHz carrier. An extensive data set including raw acoustic data source localization information, and modem position was collected during the deployment.

An additional function of the acoustic network was to communicate with, and track, the *Odyssey*, an autonomous underwater vehicle operated by the MIT group at the SIMI camp. To this end, the *Odyssey* was equipped with a Datasonics modem configured for periodic QPSK transmission to the network. A data set was obtained from which both the up-link communication and localization capabilities of the network can be determined.

17. Document Analysis a. Descriptors

acoustic modem
arctic
communication

b. Identifiers/Open-Ended Terms

c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
Approved for public release; distribution unlimited.	UNCLASSIFIED	56
	20. Security Class (This Page)	22. Price

(See ANSI-Z39.18) *See Instructions on Reverse* OPTIONAL FORM 272 (4-77)
(Formerly NTIS-35)
Department of Commerce